

⑯ 日本国特許庁 (JP)

⑮ 特許出願公開

⑰ 公開特許公報 (A)

昭55—137360

⑯ Int. Cl.³
F 02 P 5/04
F 02 D 5/02
35/00
// B. 60 R 16/02

識別記号

厅内整理番号
8011—3G
6933—3G
7604—3G
6839—3D

⑭ 公開 昭和55年(1980)10月27日
発明の数 1
審査請求 未請求

(全 6 頁)

⑯ 自動車用制御装置

⑯ 特 願 昭54—45324
⑯ 出 願 昭54(1979)4月16日
⑯ 発明者 安保敏巳

横浜市神奈川区西寺尾町714

⑯ 出願人 日産自動車株式会社
横浜市神奈川区宝町2番地
⑯ 代理人 弁理士 中村純之助

明細書

1. 発明の名称 自動車用制御装置

2. 特許請求の範囲

1. ストアド・プログラム方式のデジタル計算機を用いて車載機器を制御する自動車用制御装置において、基本的構成が同一で一部分のみが異なる複数のプログラムを一つのサブルーチンとして構成し、上記サブルーチンの実行に先立って該サブルーチンの使い方を指定することにより、一部分が異なる複数のプログラムの演算を一つのサブルーチンで共通に行なわせるように構成した自動車用制御装置。

2. 上記サブルーチンとして、内燃機関の運転条件に応じて制御量を予めセットされたデータテーブルからルックアップするプログラムすなわち、テーブル・ルックアップ・ルーチンを用い、データテーブルの格納番地及びそのデータ量を指定してからテーブル・ルックアップ・ルーチンへジャンプさせることにより、データ量の異なった複数

のデータテーブルのルックアップを一つのテーブル・ルックアップ・ルーチンで共通に行なわせることを特徴とする特許請求の範囲第1項記載の自動車用制御装置。

3. 上記サブルーチンとして、入力データの演算処理を行なうプログラムを用い、入力データの有効桁数や小数点位置を示す形式、処理後のデータの形式及び演算処理に使用する定数のうちの少なくとも一つを指定してジャンプさせることにより、形式等の異なるデータの演算を一つのサブルーチンで共通に行なわせることを特徴とする特許請求の範囲第1項記載の自動車用制御装置。

4. 上記サブルーチンとして、データに同一処理を複数回繰返して行なうプログラムを用い、繰返しの回数を指定してジャンプさせることにより、繰返し回数の異なるデータ処理を一つのサブルーチンで共通に行なわせることを特徴とする特許請求の範囲第1項記載の自動車用制御装置。

5. ジャンプする行先番地を指定することによって繰返し回数を選択することを特徴とする特許

請求の範囲第4項記載の自動車用制御装置。

3. 発明の詳細を説明

本発明はストップ・プログラム方式のデジタル計算機（いわゆるマイクロコンピュータ）を用いて各種の車載機器を制御する自動車用制御装置に関するものである。

最近、マイクロコンピュータを用いて自動車の各種車載機器、例えば内燃機関（燃料供給量、点火時期、排気還流量、回転速度等の制御）、変速機、ブレーキ、ラジオ、メータ類等を制御する装置が開発されている。

上記のごとき制御装置のプログラミングの手法として、同一の処理が何度も繰返して行なわれるものについては、サブルーチンが良く用いられる。

例えば、第1図の様なフローチャートの場合、処理A→処理J→処理B→処理J→処理Cの順番で実行されるが、これを第2図のように処理Jを、サブルーチンとし、処理Aを実行後、JSR（サブルーチンにジャンプせよ）という命令を挿入す。

• 3 •

ムをもサブルーチン化出来るようにし、プログラムの量を更に削減できるようにした自動車用制御装置を提供することを目的とする。

以下図面に基づいて本発明を詳細に説明する。

第4図は本発明を適用する自動車用制御装置の一例のブロック図である。

第4図において、1は被制御機器であり、例えば内燃機関である。また2は制御装置であり、出入力装置3、中央演算装置（CPU）4、読み出し専用メモリ（ROM）5、読み出し書き込みメモリ（RAM）6等から構成されている。

内燃機関1からは、機関運転状態を示す各種の信号（回転速度信号、吸入空気量信号、水温信号、スロットル弁全閉スイッチ等の各種スイッチの信号等）が出入力装置3を介してCPU4に送られる。CPU4はROM5で記憶されているプログラムに従って演算を行ない、各種の制御信号（燃料供給量、点火遅角、排気還流率等を制御する信号）を出入力装置3を介して内燃機関1に送って、所望の制御を行なう。

れば処理Jを実行し、RTS（サブルーチンからリターンせよ）という命令でもとに戻り、続いて処理Bを実行する。再びJSRにより処理Jを実行し、RTSで処理Cを実行する。これにより処理Jを2回プログラムする必要はなくなり、プログラムの量を削減することが出来るので、メモリを節約することが出来る。

しかし、上記のごとき従来のサブルーチンにおいては、全く同一のプログラムはサブルーチン化出来るが、プログラムの構成は良く似ていても、その一部（扱うデータのビット数、扱うデータの量、繰返し処理の回数等）が異なる場合には、サブルーチン化することが出来なかつた。例えば、第3図のJとJ'の構成が非常に似たものであっても、データのビット数等が異なるだけでサブルーチン化することが出来ないという問題があった。

本発明は上記の問題に鑑みてなされたものであり、サブルーチンにジャンプする前にそのサブルーチンの使い方を指定してからジャンプさせるよう構成することにより、一部が異なるプログラ

• 4 •

次に第5図は本発明の一実施例図であり、(1)はフローチャート、(2)はアクチュエータを示す。

第5図の実施例は、8ビットのマイクロコンピュータ（例えばモトローラ6800）において、CPU内のアクチュエータA、B（図ではAcc.A及びAcc.Bと表示、各々8ビット）の内容を二つ組合せて16ビットのデータとして用いる場合に、アクチュエータA、Bの内容のシフトの回数を選択できるサブルーチンを示す。

すなわち、アクチュエータの内容を右へシフトするルーチンR₁、R₂、R₃を縦に接続し、ジャンプする位置を指定することによって、シフトの回数を選択出来るようにしたものであり、例えばSHIFT3へジャンプすれば、サブルーチンが終了（RTS命令まで）するまでにシフトを3回行なう。同様にSHIFT2、SHIFT1へジャンプすれば、それぞれ2回、1回シフトされる。

したがって第5図のサブルーチンを用い、ジャンプする位置を指定してやれば、一つのサブルーチンで異なったシフト回数の演算を自由に選択し

• 5 •

• 6 •

て行なわせることが出来るので、シフト回数の異なった複数のプログラムを備えておく必要がなくなる。

次に、第6図は本発明の第2の実施例のフローチャートである。

第6図の実施例は、例えばA/D変換された入力データに演算処理を行なうサブルーチン（サブルーチン名はCAL）で、定数+A/D変換データという計算を実行して、演算結果を8ビット精度で出す場合と10ビット精度で出す場合とを一つの計算ルーチンで行なうものを示す。

第6図(イ)において、8ビット精度データをA/D変換すると、FLAG(FLAGという名前を付けたRAM(メモリ))を0にしてCALへジャンプする。また、第6図(ロ)において10ビット精度データをA/D変換するとFLAGを1にしてCALへジャンプする。次に第6図(ハ)において、サブルーチンCALではその実行に先立つて入力データが10ビット精度か8ビット精度かをFLAGによって判断し、1.0ビット精度

(FLAG=1)の場合、被除数である定数を2ビット左ヘシフトして8ビット精度の場合と条件を同じにして計算を実行すれば、同一の計算ルーチンで両方を実行することが出来る。

上記のようなデータの形式(有効桁数や小数点位置)や、また更に演算処理に使用する定数が異なる場合でもサブルーチン化することが出来る。

次に、第7図は本発明の第3の実施例のフローチャートである。

この実施例は、エンジンの運転条件(回転数、負荷量)に応じて2次元テーブルから最適制御量(進角、燃料補正量)をルックアップするサブルーチン(テーブル・ルックアップ・ルーチン)である。まず実施例の説明に先立って、テーブル・ルックアップ・ルーチンについて、第8図に基づいて説明する。第8図(イ)は、エンジン回転数と負荷量から点火進角をルックアップするデータテーブルの1例図、(ロ)は、これをROMにストアした例を示す図である。(イ)のデータテーブルの横軸N₁～N₄は例えばエンジン回転数であり縦軸L₁～

7

8

～L₈は負荷量である。今エンジン回転数NがN₁ < N < N₄で負荷量LがL₁ < L < L₈であったとする、それぞれに対応するデータD₁₁、D₁₂、D₁₃、D₁₄を用いて補間計算をすることにより回転数N、負荷量Lに対応する進角値Dが計算できる。

実際の手順は、計測された負荷量LをもとにしてBEBB番地(Bは16進法の14、Bは16進法を示す)から順に格納してあるL₁～L₈を使ってL₁ < L < L₈となることを判別し、続いて、回転数NをもとにしてBEB2番地から順に格納してあるN₁～N₄を使ってN₁ < N < N₄となることを判別し、この二つの結果から、BEB4番地から格納してあるメインテーブルのD₁₁、D₁₂、D₁₃、D₁₄を使って補間計算を行なう。本例は、回転数、負荷量を、共に8分割した8×8のテーブルについて行なったが、例えば16分割の16×16のテーブルについても同様である。

次に第7図(ロ)において、点火時期を演算するルーチン(ADV)は、回転数(8分割)、負荷量(8分割)に応じてデータテーブルから進角値

をルックアップする。この場合、8×8テーブルのルックアップを示すため、SIZE(SIZEという名前を付けたRAM(メモリ))を0にしておく。また点火進角の場合、トランスマッシュの形式に応じて点火進角を変えるため、データテーブルの格納番地が異なっている。このためMT(マニュアルトランスマッシュ)かAT(オートマチックトランスマッシュ)かに応じて、データテーブルの格納番地をT₁、T₂、T₃(それぞれ名前を付けたRAM)にセットしてLOOKUPルーチンへジャンプするようにしておく。また第7図(ハ)において、燃料補正量を演算するルーチンは、回転数、負荷量共16分割で精度を上げて16×16のテーブルルックアップを行なう。これを示すためSIZEを1にしてさらにデータテーブルの格納番地をT₁T₂T₃にセットしてLOOKUPルーチンへジャンプする。

T₁、T₂、T₃のセットは、具体的には第7図(ニ)に示す様に、メモリ(ROM)内に縦軸格子点データS₁(例えば前記L₁～L₈)、横軸格子

9

10

点データ S_i (例えば前記 $N_1 \sim N_8$)、メインテーブル S_8 (8×8 あるいは 16×16) が格納されているとすると、 $T_1 \rightarrow S_8 S_8 S_8$ 、 $T_2 \rightarrow S_8 S_8 S_8$ 、 $T_3 \rightarrow S_8 S_8 S_8$ (但し S は 1.6 進法を示す) をセットすることになる。

次に第 7 図 (I) の LOOK UP ルーチンでは、SIZE をチェックして 8×8 のテーブル・ルックアップか 16×16 のテーブル・ルックアップかを判断しそれに応じてテーブル・ルックアップを行なう。

上記のように第 7 図の実施例においては、 8×8 のテーブル・ルックアップと、精度の高い 16×16 のテーブル・ルックアップとを一つのサブルーチンで行なうことが出来る。

以上説明したごとく本発明においては、サブルーチンの使い方を指定してからジャンプさせることにより、構成の似たルーチンは全てまとめて一つのサブルーチンにすることが出来るので、プログラムの量、したがってメモリの量を大巾に節約することが出来るという効果がある。

4. 図面の簡単な説明

第 1 図～第 3 図は従来例のフローチャート、第 4 図は本発明を適用する自動車用制御装置の一例図、第 5 図～第 7 図はそれぞれ本発明の実施例のフローチャート、第 8 図はデータテーブルの一例図である。

符号の説明

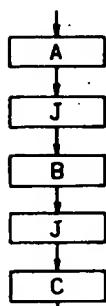
1 … 被制御機器	2 … 制御装置
3 … 入出力装置	4 … C P U
5 … R O M	6 … R A M

代理人弁理士 中 村 純之助

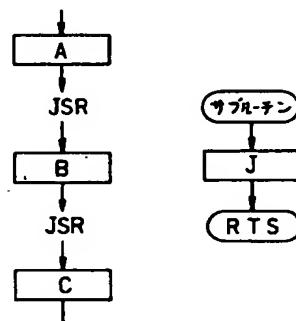
- 11 -

- 12 -

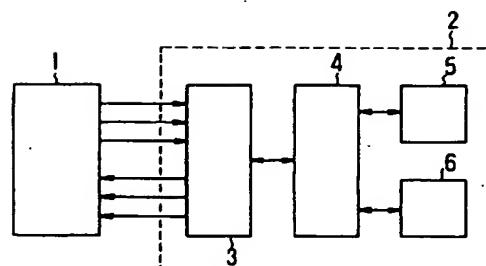
オ 1 図



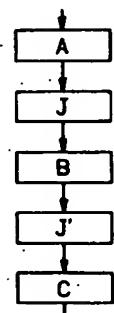
オ 2 図



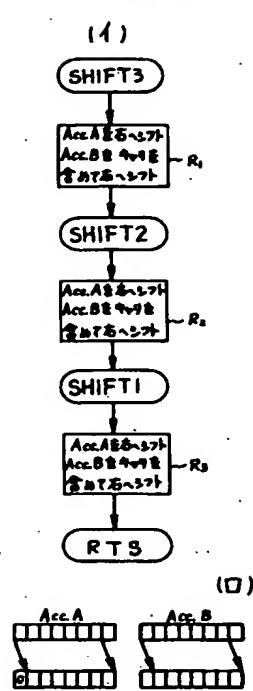
オ 4 図



オ 3 図



第 5 回



第6圖

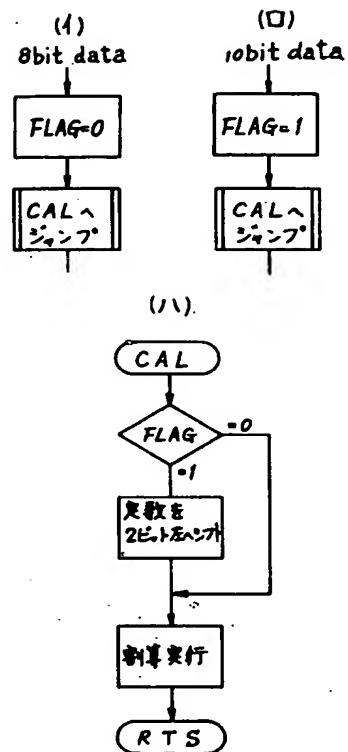
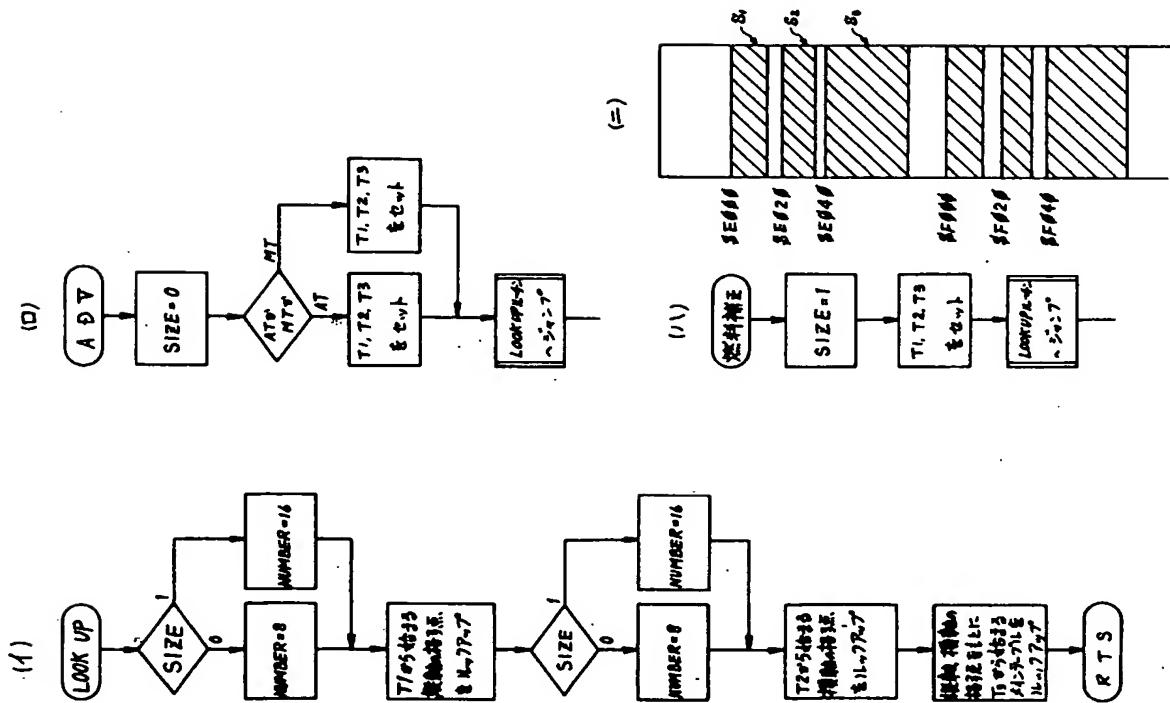


圖
七
赤



第8回

(1)

L_0	D_{10}	D_{20}	D_{30}	D_{40}	D_{50}	D_{60}	D_{70}	D_{80}
L_1	D_{11}	D_{21}	D_{31}	D_{41}	D_{51}	D_{61}	D_{71}	D_{81}
L_2	D_{12}	D_{22}	D_{32}	D_{42}	D_{52}	D_{62}	D_{72}	D_{82}
L_3	D_{13}	D_{23}	D_{33}	D_{43}	D_{53}	D_{63}	D_{73}	D_{83}
L_4	D_{14}	D_{24}	D_{34}	D_{44}	D_{54}	D_{64}	D_{74}	D_{84}
L_5	D_{15}	D_{25}	D_{35}	D_{45}	D_{55}	D_{65}	D_{75}	D_{85}
L_6	D_{16}	D_{26}	D_{36}	D_{46}	D_{56}	D_{66}	D_{76}	D_{86}
L_7	D_{17}	D_{27}	D_{37}	D_{47}	D_{57}	D_{67}	D_{77}	D_{87}
L_8	D_{18}	D_{28}	D_{38}	D_{48}	D_{58}	D_{68}	D_{78}	D_{88}
L_9	D_{19}	D_{29}	D_{39}	D_{49}	D_{59}	D_{69}	D_{79}	D_{89}
L_{10}	D_{10}	D_{20}	D_{30}	D_{40}	D_{50}	D_{60}	D_{70}	D_{80}
L_{11}	D_{11}	D_{21}	D_{31}	D_{41}	D_{51}	D_{61}	D_{71}	D_{81}
L_{12}	D_{12}	D_{22}	D_{32}	D_{42}	D_{52}	D_{62}	D_{72}	D_{82}
L_{13}	D_{13}	D_{23}	D_{33}	D_{43}	D_{53}	D_{63}	D_{73}	D_{83}
L_{14}	D_{14}	D_{24}	D_{34}	D_{44}	D_{54}	D_{64}	D_{74}	D_{84}
L_{15}	D_{15}	D_{25}	D_{35}	D_{45}	D_{55}	D_{65}	D_{75}	D_{85}
L_{16}	D_{16}	D_{26}	D_{36}	D_{46}	D_{56}	D_{66}	D_{76}	D_{86}
L_{17}	D_{17}	D_{27}	D_{37}	D_{47}	D_{57}	D_{67}	D_{77}	D_{87}
L_{18}	D_{18}	D_{28}	D_{38}	D_{48}	D_{58}	D_{68}	D_{78}	D_{88}
L_{19}	D_{19}	D_{29}	D_{39}	D_{49}	D_{59}	D_{69}	D_{79}	D_{89}
L_{20}	D_{10}	D_{20}	D_{30}	D_{40}	D_{50}	D_{60}	D_{70}	D_{80}

(□)

(ア) リズム

SE#80

SE#20

SE#40

SE#50

SE#60

SE#70

L_1	L_2	L_3	L_4	L_5	L_6	L_7	L_8
N_1	N_2	N_3	N_4	N_5	N_6	N_7	N_8
D_{10}	D_{20}	D_{30}	D_{40}	D_{50}	D_{60}	D_{70}	D_{80}
D_{11}	D_{21}	D_{31}	D_{41}	D_{51}	D_{61}	D_{71}	D_{81}
D_{12}	D_{22}	D_{32}	D_{42}	D_{52}	D_{62}	D_{72}	D_{82}
D_{13}	D_{23}	D_{33}	D_{43}	D_{53}	D_{63}	D_{73}	D_{83}
D_{14}	D_{24}	D_{34}	D_{44}	D_{54}	D_{64}	D_{74}	D_{84}
D_{15}	D_{25}	D_{35}	D_{45}	D_{55}	D_{65}	D_{75}	D_{85}
D_{16}	D_{26}	D_{36}	D_{46}	D_{56}	D_{66}	D_{76}	D_{86}
D_{17}	D_{27}	D_{37}	D_{47}	D_{57}	D_{67}	D_{77}	D_{87}
D_{18}	D_{28}	D_{38}	D_{48}	D_{58}	D_{68}	D_{78}	D_{88}
D_{19}	D_{29}	D_{39}	D_{49}	D_{59}	D_{69}	D_{79}	D_{89}

(b)